snom 4S NAT Filter Admin Manual

snom
VoIP-phones

snom 4S
NAT Filter
Version 2.11

snom 4S NAT Filter Version 2.11

Table of Contents

1 Overview

Network address translation (NAT) is a reality today. There have been many discussions about the evil and the good of this network topology and the replacement by IP version 6. However, operators and business want to offer VoIP services today and therefore must address the problem.

The snom 4S NAT Filter is a SIP session border controller (SBC). It enables non-NAT aware devices to operate in private networks. It also allows operating the data center in a private network. It takes care about translation of SIP messages with private network identities into identities that can be addressed from the data center. When necessary or explicitly required, it forwards media and changes the SDP attachments in the SIP messages accordingly.

The SBC offers recording capabilities (depending on the licensing). Through a separate management interface, operators can define numbers and patterns that are silently recorded. Users may explicitly request the recording of a call by pressing a key on the phone; in this case the whole call will be recorded (even parts of the call before pressing the key). The filter records in a compressed format where only the voice part of the conversation is recorded in highly-compressed audio format (13.2 kBit/s). By using an appropriately configured web server, operators can manage very large numbers of calls and, for example, forward them per e-mail to users or authorities.

The SBC does not translate signaling, e.g., from SIP to H.323 or to MGCP. It is a semi-transparent SIP proxy that takes care only about known methods and leaves the rest unchanged. If all user agents are fully NAT-compliant or on public Internet, the filter can transparently be removed from the network without changing the call flows or functionality. Also, the filter does not interfere with unknown applications. This is a advantage over back-to-back user agents (B2BUA) that operate on the application level.

1.1 Applications

The filter can be used in the following scenarios:

- Corporations. Corporations which operate their infrastructure behind NAT and/or firewalls can talk to the public Internet through the filter.
- Operators. Operators offer the NAT traversal feature to their customers. Using the scalability feature of the filter, the operation of large networks becomes possible.
- Record specific calls for legal purposes. In many countries, operators must provide the possibility to record certain calls on request. The filter can perform this task.
- Recording can be used for legal proofing (brokers, etc). The filter is fully compliant with other SIP equipment and can, for example, put between a PSTN gateway and SIP phones.

1.2 Features

The filter offers powerful features based on modern VoIP technology:

- The built-in RFC3261-compliant SIP proxy makes additional external SIP logic superfluous and simplifies the system setup.
- A built-in RFC3489-compatible STUN server for single IP addresses allows clients to self-refresh their bindings
- Support for instant messaging, presence and all other SIP-compliant applications.
- Rich logging features allow easy maintenance.
- Recording functions based on number lists and expressions offer a flexible way of filtering out information.
- Recordings can be saved in WAV file format (the data rate is 6 MB per hour).
- Almost stateless operation allows the filter to be used in server farms. This offers a tremendous scalability and redundancy making the product suitable for large operators.

- Both http and https as web interface for simple access from anywhere on the Internet.
- The filter supports Interactive Connectivity Establishment (ICE). User agents that support this feature will optimize the media path for the shortest possible delay.
- Media relay is established using connection-oriented media. User-agents that are not NAT-aware inherently support this feature. This makes the operation of the NAT filter backward-compatible.
- Call-alive polling. During calls, the filter checks if the call is still alive and terminates the call if this should not be the case. With this feature, charging users for broken calls can be avoided.
- Reliable and unreliable transport layers. The filter supports both UDP and TCP transport layers. Full TLS support will be added soon.
- To and From headers may be changed for calls. The filter talks to a web application server to get this information.
- Application Server Integration. The web application server can also change the request-URI. This makes simple routing possible, which can be used for least cost routing, for example.
- Call Duration Limit. The web application server can define the duration of the call. This makes it possible to implement prepaid services.
- HTTP Registrar. The SBC may convert SIP register requests into HTTP requests and send the response back via SIP. This makes it possible to use the web server as registrar.
- Authentication. The SBC may authenticate incoming requests. It keeps a authentication cache updated by the web server. Trust relationships defined by IP address define exceptions.
- CLIR support. When a request leaves the data center, the SBC can change the From-header of the outgoing request and make it anonymous. This way, the caller ID can be hidden.

Usually, the filter acts as stateless proxy. That means, by default it just forwards the packets and does not change the content of the attachments or the headers themselves. That means that the filter will not interfere with applications (instant messaging, presence, weather report, etc).

There are three exceptions to this rule:

- The first exception is a REGISTER request. When a user agent tries to register and needs the support of the filter, the filter will set up a local data structure representing the user agents. It will make sure that the connection to the user agents stays alive. It will also make sure that requests destined to the user agents will be forwarded properly.
- The second exception is an SDP attachment. The filter checks if the user agent needs support (or must be recorded) and will in that case add a local contact to the SDP that can be used for media relay.
- The third exception occurs when the filter queries a web server for routing information. In this case, it will send a provisional response to stop the UAC from repeating messages.

These three exceptions make sure that all user agents will work behind NAT, no matter what NAT-type or how many NAT-levels are being used. If user agents support ICE, they will automatically find the shortest media path to the other party (peer-to-peer).

2 Architecture

2.1 The NAT Filter and SIP

In the SIP architecture, the SBC acts as the first proxy that is contacted by user agents. There are two ways to make sure that the relevant traffic gets routed trough the filter:

- User agents can be set up to use the filter as outbound proxy. When using this method, all SIP traffic will flow through the SBC, whether it is destined to the operator or not. That means that service for calls outside of the operator's domain may also be serviced by the SBC. However, by redirecting all outgoing traffic of the SBC to a proxy the operator can make sure that the authentication, authorization and accounting (AAA) requirements for requiring the service are fulfilled. Alternatively, you can use the application server interface to do the job on the SBC itself.
- User agents resolve the SBC though the RFC3263 DNS resolving process. That means that only the traffic that is destined to the operator's domain will use the service of the NAT Filter. However, users might be annoyed if they place a call to a domain that does not properly support NAT services. In this case, the SBC can also redirect the traffic to another proxy.

We recommend using the first alternative and to only choose the second alternative if it is too difficult to provision user agents with the outbound proxy or when there are concerns about providing service for foreign operators.

Usually, the SBC acts as stateless proxy. This means, that it just forwards the packets by default and that it does not change the content of the attachments or the headers themselves. The SBC will not interfere with applications (instant messaging, presence, weather report, etc).

There are three exceptions to this rule:

- The first exception is a REGISTER request. When a user agent tries

to register and needs the support of the SBC, the SBC will set up a local data structure representing the user agents. It will make sure that the connection to the user agents stays alive. It will also make sure that requests destined to the user agents will be forwarded properly.

- The second exception is an SDP attachment. The SBC checks if the user agent needs support (or must be recorded) and, in that case, will add a local contact to the SDP that can be used for media relay.
- The third exception occurs when the SBC queries a web server for routing information. In this case, it will send a provisional response to stop the UAC from repeating messages.

These three exceptions make sure that all user agents will work behind NAT, no matter what NAT-type or how many NAT-levels are being used. If user agents support ICE, they will automatically find the shortest media path to the other party (peer-to-peer).

2.2 NAT

Network Address Translation (NAT) is a reality in today's networks. Many operators save IP addresses by providing only one IP address for a number of devices, sometimes companies. Firewall manufacturers make NAT a feature by performing inspection of packets that go though NAT. Even for IPv6 networks, the fundamental problem will remain as there will also be a need for firewalls and private networks.

The Session Initiation Protocol (SIP) has neglected this problem in the beginning. However, in some recent RFC there have been useful proposals on how to deal with the problem. This document shows how the snom 4S NAT Filter can be used to solve the problems.

Although snom also makes user agents, the snom 4S NAT Filter works with most SIP user agents from other companies. The requirements on these user agents are described below.

If you want to use the SBC just for recording purposes, you don't need to bother about NAT. The SBC also works when no NAT is present.

2.2.1 How does NAT work?

NAT is essentially a translation table that maps public IP address and ports combinations to private IP address and port combinations.

The translation table is implicitly set up when a packet is sent from the private network to the public network. The association is kept alive for a certain time and is refreshed every time a new packet is sent from the same origin. This fact is used by STUN (RFC3489) to set up an association between a public IP address and a private IP address.

In symmetrical NAT, the router stores the address where the packet was sent. Only packets coming from this address are forwarded to the private address. This algorithm increases the security as it is harder to guess the source IP and port for attackers. Full cone NAT does not perform this check.

There are some mixed variants between full cone NAT and symmetrical NAT. Restricted port NAT works similar to symmetrical NAT, but uses only one port association.

Hairpinning is the ability of the NAT to route packets coming from the private network and addressed towards a public IP address binding back to the private network. Not all routers support this feature.

2.2.2 Symmetrical RTP

Real time protocol (RTP) is used to transport media. Symmetrical RTP is a trick to extend the number of cases when communication can be established. A SIP user agent supporting symmetrical RTP waits for the first RTP packet coming in and then sends its media stream back to the IP address from which it received that packet. Symmetrical RTP always works when the user agent doing symmetrical RTP is on a globally routable address. However, this algorithm can easily be cheated (port spraying) and therefore implies a certain security risk.

2.2.3 Signalling SIP

SIP traffic is relatively unproblematic because SIP typically is not as time critical as media. Usually, it is ok to route SIP packets through a longer path than media.

In SIP it is legal to send from a different port than the receiving port. When this is being done, there is no way of supporting these devices behind NAT. However, some phones offer an option that disables this mechanism so that the sending port is the same as the receiving port.

Typically, the SIP proxy will run on a public IP address where it is possible to deal with all kinds of NAT. Keep-Alive messages may keep the NAT binding open (for example, short registration periods or non-SIP messages).

2.2.4 Media RTP

Media is much more problematic than SIP because users are sensitive to delay in a voice conversation. When the delay is too long, the speakers need to be disciplined not to interrupt the other person when starting to speak. Also, the ear is much more sensitive to echo when the media delay becomes too long. The effect is known from intercontinental calls where the speed of light increases the delay for voice transmission.

SIP was designed for peer-to-peer communication. That means the user agents (telephones) send the media directly to the other user agent. This approach is the best way to minimize the delay; however, it becomes a problem when NAT is involved.

2.2.5 Classification of User Agents

From a SBC point of view, available user agents can be classified into the following categories:

- Public IP devices. These devices operate on public IP addresses and don’t need any specific support regarding NAT. The true location of these devices may be in a private network, as they might have allocated a public identity using mechanisms like UPnP™ [3]. These devices are most welcome as they don’t cause any additional requirements.
- STUN devices. Phones that operate behind full cone NAT and allocate public IP addresses themselves fall into this category. The only support that the proxy needs to give is a STUN server. Apart from that they act like public IP devices.
- Non NAT-aware devices. These devices don’t attempt to check the NAT type or to allocate a public IP address. Often, they are “legacy”

devices that have been designed without having NAT in mind. These devices can register only for a short period of time, so that the REGISTER messages keep the port association open (the SIP messages are used to keep the port association). Also, these devices need a NAT-aware media server or other device that forward the RTP packets of these devices.

- Symmetrical NAT devices. These devices may be NAT-aware; however, because they operate behind symmetrical NAT, there is little that they can do. They essentially behave like non NAT-aware SIP devices and hope for the support of the proxy.

2.2.6 Probing Media Paths

ICE is a method that has been proposed recently in the IETF [4]. The algorithm is simple: A user agent that supports ICE lists the possible addresses where it could possibly be reached. These addresses may include the private address, an address allocated via STUN, one or more addresses allocated with the TURN protocol or an address allocated with UPnP. Because in practice it is hard to predict which of these addresses are visible to the other user agent, all of the possible addresses are proposed to the other user agent.

The other user agent sends test packets to the possible addresses. Picking the first reply on the test packet will establish a working media path and it will also probably be the fastest connection. STUN is being used for these test packets.

2.2.7 The Role of the NAT Filter

When a user agent is not able to allocate a globally routable address or it is not sure if it found enough possible addresses, the NAT Filter can help out.

Again, the way the NAT Filter works is simple. For the signalling, the NAT Filter keeps the NAT alive with bogus messages (which can be SIP messages or other non-SIP message). It patches the messages in such a way that other user agents will address the NAT Filter instead of the user agent when they want to deliver a message. The NAT Filter then forwards the message to the user agent using the connection which is kept open with the keep-alive messages.

When the NAT Filter sees a message that contains information about sending media (session description protocol, SDP), it opens a local globally routable port on behalf of the user agent and patches these messages in a way that the destination will send media via this port. The NAT Filter will relay the media to the user agent like it relays SIP messages. Using symmetrical RTP, it can detect the user agent's public media identity and reroute the packets to this destination.

While this approach has the huge advantage that it works with all kinds of NAT, it has the disadvantage that it increases the media path significantly. For example, when a user A in Tokyo is registered with an operator in New York and wants to call his colleague B (who is registered with a service provider in Sydney and sitting in the same office in a private network), the media would have to flow first from Tokyo to New York, then to Sydney and then back to Tokyo. The delay would at least be around one second even though the user agents are located in the same network.

Unfortunately, it is not trivial to make the media path shorter. There have been some attempts to reduce the problem, but it is much easier to address the problem starting at the user agent. If the user agent uses ICE, it will try all addresses listed in the SDP attachment, including the port allocated by the NAT Filter. If there is a shorter path, it will switch to this shorter path. If there is no other way or if the other side does not support ICE, it will fall back to the NAT Filter-allocated port which will work in all cases.

2.2.8 Optimizing the Media Path for Symmetrical NAT

In the cases where both user agents are behind symmetrical NAT, the NAT Filter approach will ensure that media will flow between the user agents. However, the Tokyo example shows that this might result in intolerable media delay.

To address this problem, TURN [5] comes into play. The idea behind this approach is to allocate identities on several places in the Internet and to propose all of the allocated ports to the other user agent. If the ports are allocated on all continents, the other user agent will automatically pick the TURN server with the shortest delay. In the Tokyo example, a TURN server located in Japan will reduce the delay to a tolerable level (even when there is no direct path between the user agents).

2.3 SBC Behaviour

2.3.1 Registering

When a user agent registers, it puts its IP address in the top Via. If the user agent is on public Internet or properly supports NAT, this Via will match the perceived IP address. In this case the SBC does not interfere with the registering process and just forwards this packet to the registrar.

If the top Via does not contain the perceived address, the SBC will take care of the request. It will replace the provided contact with a locally generated contact and forward the request to the registrar (see below).

```
REGISTER sip:snomag.de SIP/2.0
Via: SIP/2.0/UDP 203.145.183.113:12975;branch=z9hG4bK-
abx3au3mxb01;rport
From: <sip:denny@snomag.de>;tag=k9p6fmeg7h
To: <sip:denny@snomag.de>
Call-ID: 3c26701d7cb9-pady07b5783t@203-145-183-113
CSeq: 14 REGISTER
Max-Forwards: 70
Contact: <sip:denny@203.145.183.113:12975;line=lhynyb3y>;q=1.0
Supported: gruu
Expires: 86400
Content-Length: 0

REGISTER sip:snomag.de SIP/2.0
Via: SIP/2.0/UDP 217.115.141.99:5082;branch=z9hG4bK-e8d1feb8138c3d85
0637ced821ef40a3;ua=c9b140ab598290e5bb491e9c3aaca440
Via: SIP/2.0/UDP 203.145.183.113:12975;branch=z9hG4bK-
abx3au3mxb01;rport=17401
From: <sip:denny@snomag.de>;tag=k9p6fmeg7h
To: <sip:denny@snomag.de>
Call-ID: 3c26701d7cb9-pady07b5783t@203-145-183-113
CSeq: 14 REGISTER
Max-Forwards: 69
Contact: <sip:217.115.141.99:5082;ua=c9b140ab59829bb491e9c3aaca440>
Supported: gruu
Expires: 86400
Content-Length: 0

SIP/2.0 200 Ok
Via: SIP/2.0/UDP 217.115.141.99:5082;branch=z9hG4bK-e8d1feb8138c3d85
```

```
0637ced821ef40a3;ua=c9b140ab598290e5bb491e9c3aaca440
Via: SIP/2.0/UDP 203.145.183.113:12975;branch=z9hG4bK-
abx3au3mxb01;rport=17401
From: <sip:denny@snomag.de>;tag=k9p6fmeg7h
To: <sip:denny@snomag.de>;tag=epuy85kzm5
Call-ID: 3c26701d7cb9-pady07b5783t@203-145-183-113
CSeq: 14 REGISTER
Contact: <sip:217.115.141.99:5082;ua=c9b140ab598290e5bb491e9c3aaca44
0>;expires=3600;gruu="sip:denny@snomag.de;gruu=hobiv52b"
Date: Wed, 26 May 2004 16:03:33 GMT
Content-Length: 0

SIP/2.0 200 Ok
Via: SIP/2.0/UDP 203.145.183.113:12975;branch=z9hG4bK-
abx3au3mxb01;rport=17401
From: "denny" <sip:denny@snomag.de>;tag=k9p6fmeg7h
To: "denny" <sip:denny@snomag.de>;tag=epuy85kzm5
Call-ID: 3c26701d7cb9-pady07b5783t@203-145-183-113
CSeq: 14 REGISTER
Contact: <sip:denny@203.145.183.113:12975;line=lhynyb3y>;expires=360
0;gruu="sip:denny@snomag.de;gruu=hobiv52b"
Date: Wed, 26 May 2004 16:03:33 GMT
Content-Length: 0
```

The SBC will make the registration short enough to keep the connection alive. The UA will reregister shortly after. However, because the registration binding time in the registrar is longer, the SBC will not forward the request to the registrar and answer it locally.

2.3.3 RTP Relay

When initiating a call, user agents usually include a Session Description Protocol (SDP) attachment that describes where they expect media. If the user agent operates on a public Internet address, there is no need to interfere in this process. In this case the SBC will just forward the request.

Operators should encourage customers to use equipment that operates on a public Internet address or properly allocates a globally routable Internet address. Because media relay is an expensive operation, it reduces the overall load on the network and at the same time increases the quality of the service.

However, when a user agent is behind NAT, it might not be able to receive media directly. In some cases this is because the user agent is

simply not programmed to allocate an address properly or because it is behind symmetrical NAT, which makes it impossible to properly allocate this address. In this case, the help of the media SBC will make sure that media will always be delivered properly.

The media filter supports the "interactive connectivity establishment" (ICE) method that has been published recently in the IETF. Using this method, user agents may probe several addresses and decide which address they use for communication. In this case, the SBC will just add another contact to the ICE list.

Table 1 shows the cases when the SBC needs to interfere if STUN and ICE support are available from the user agents. The support of the SBC is necessary only in cases when both sides have symmetrical NAT and in the case when talking from symmetrical NAT to restricted NAT. If the user agents don't support STUN and ICE, the number of cases goes up significantly.

If the user agent operates without NAT support, it will send a SDP like the one below:n

```
v=0
o=root 19387 19387 IN IP4 192.168.1.10
s=call
c=IN IP4 192.168.1.10
t=0 0
m=audio 58146 RTP/AVP 0 8 3 18 2 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:18 g729/8000
a=rtpmap:2 g726-32/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
```

The NAT Filter will detect that the user agents needs help and allocates local ports for relaying media. It will forward the request with changed SDP:

```
v=0
o=root 19387 19387 IN IP4 217.115.141.99
s=call
c=IN IP4 217.115.141.99
t=0 0
m=audio 49170 RTP/AVP 0 8 3 18 2 101
```

```
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:18 g729/8000
a=rtpmap:2 g726-32/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
a=silenceSupp:off - - - -
```

The NAT Filter changes the private address to a globally routable address and inserts the local port. It also inserts a hint that tells the other user agent that it should not do silence suppression. This reduces the risk that the connection is closed during a talk spurt of one of the parties.

2.4 Scaling and Redundancy

The NAT Filter product was designed to support distributed server farms. That means that the servers act stateless in principle; user agents may randomly pick a server in a server farm. This feature allows operators to set up very large and robust networks.

Table 1: NAT conbimations

To/From	Public IP	Full Cone NAT	Restricted NAT	Symmetrical NAT
Public IP	Always	When STUN support is available	When STUN and ICE are available	When STUN and ICE are available
Full Cone NAT		When STUN is available	When STUN and ICE are available	When STUN and ICE are available
Restricted NAT			When STUN and ICE are available	When STUN and ICE are available and no port checking
Symmetrical NAT				Needs filter

The distribution of user agents to a server is performed using DNS SRV (RFC 2782). This means that you need to list the available servers on DNS level; the user agents must perform DNS SRV look ups and pick one of the servers (possible using the detection algorithms described below).

The following table shows an example configuration for Linux named(8):

```
_sip._udp       IN SRV  3 5 5082 frankfurt1
_sip._udp       IN SRV  3 5 5082 newyork1
_sip._udp       IN SRV  3 5 5082 newyork2
_sip._udp       IN SRV  3 5 5082 newyork3
_sip._udp       IN SRV  3 5 5082 tokyo2
_sip._udp       IN SRV  3 5 5082 tokyo1
```

If one of the servers should become unavailable, the SRV algorithm will make sure that the other servers will be contacted. The user agents that are refreshed by that particular server will become unreachable until the user agents initiate a new REGISTER request. Therefore, you should make sure that your servers have a high uptime probability and that the registration period is not too long. We think that registration periods of thirty minutes up to one hour are a good balance between service failure time and performance.

2.5 Detecting the right NAT Filter

User agents must detect which server in the server farm is nearest to the user agent. This is an important feature for a company or operator that has user agents scattered around the globe. Example: A company has offices in Berlin, Tokyo and Dallas and locally operates NAT Filter servers. When a user agent is located in Tokyo, it should use the Tokyo server. This could be set up manually; howeve, it is also possible to automatically pick the best server.

To detect the nearest server, the user agent sends STUN packets to all possible servers (the servers with the lowest priority in the SRV list). The user agent picks the server that responds first. Alternatively, the user agent could send more test packets and take the mean response time for making the decision.

The snom 4S NAT Filter includes a STUN server that operates on the SIP UDP port. User agents should send their test packets to the SIP port.

2.6 Requirements on User Agents

Generally, there are two categories of user agents: The non NAT aware user agents and the STUN/ICE capable user agents.

2.6.1 Non NAT-Aware User Agents

Non-NAT aware user agents must have at least the following features:

1. Must send SIP UDP packets from the port where they receive SIP traffic
2. Must start sending media immediately after receiving SDP

Requirement 1 is not fulfilled by the default configuration of the Cisco 7960; however there is a setting that enables this feature. Most known user agents support this feature, however.

Requirement 2 sometimes creates problems with user agents who don't send media during silent periods. In this case, users have to start speaking before two-way audio can be established.

In any case, customers are asked to contact their vendor in case of problems and explanations. In general, snom recommends using NAT-aware user agents to reduce the network and support overhead.

2.6.2 STUN/ICE-Aware User Agents

STUN/ICE-Aware User Agents must implement the two IETF standards. It is ok if the user agents use the built-in STUN server for refreshing the bindings and learning the public IP address.

snom phones starting with version 2.05a fall into this category.

2.7 Defining the Maximum Session Time

There are a couple of timeout-related settings that terminate a call when certain events fire (see below). However, when prepaid cards are being used, operators want to limit the call duration to a certain time.

The SBC has a mechanism to terminate calls anyway. It does not only send BYE messages to both sides of the call, it also cuts media relaying which in practice will be used in most cases when the call is terminated via PSTN. This feature can be used to tear down calls when a card expires.

The remaining call duration depends not on a static setting, but on a dynamically provisioned parameter. This parameter is usually provided in the AAA procedure in the proxy. The proxy needs a simple way to tell the SBC how many seconds this call can stay up.

We decided to add a proprietary header called "P-Session-Timeout" to the SBC. When this SBC is detected in a message that belongs to an existing call, the SBC sets the timeout for this call to the value provided in the header (in seconds). After this time the SBC will terminate the call with the reason "Maximum Session Duration" (see below, Call History). Additionally, this parameter can be passed from the application server.

With the setting for trusted IP addresses, the SBC will accept these headers only from explicitly listed addresses. After the header has been used, the SBC removes it from the packet so that the user agent will not see this header.

If the proxy wants to provide information about how long the call can stay up, it should use AOC information.

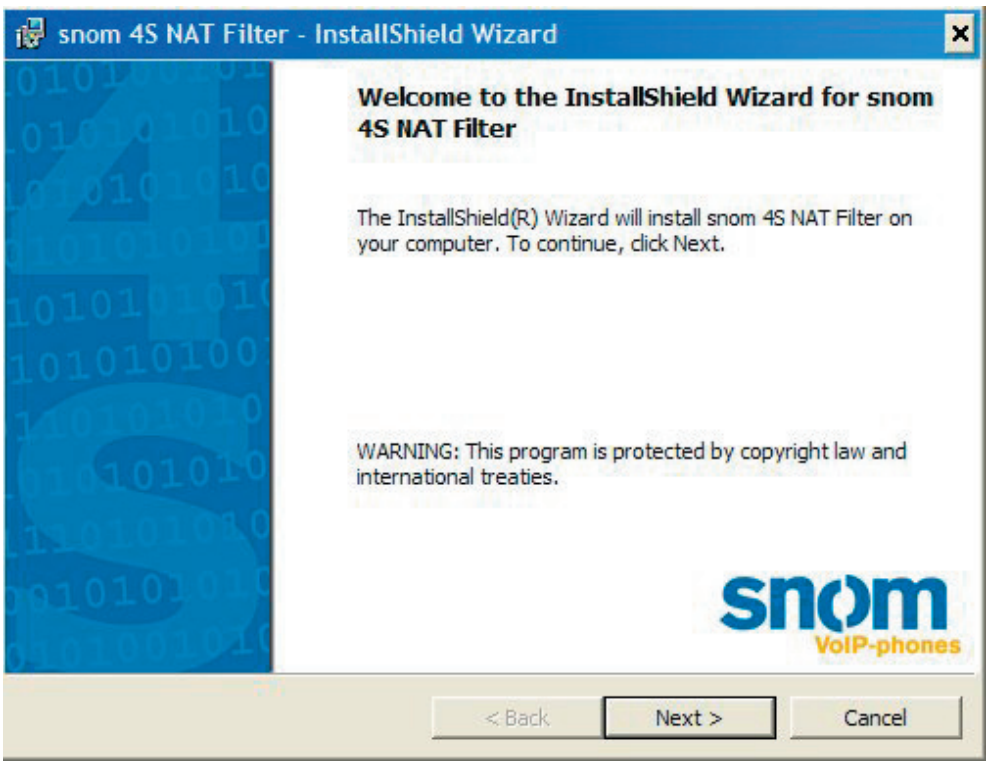
3 Installation

If you want to install the product on your own, this chapter will provide you with the necessary information.

3.1 Windows

The Windows version of the NAT Filter comes with an InstallShield application that should make the installation very simple for you.

Before you start the installation, you might want to make sure that the necessary ports are available on your machine. Please use the



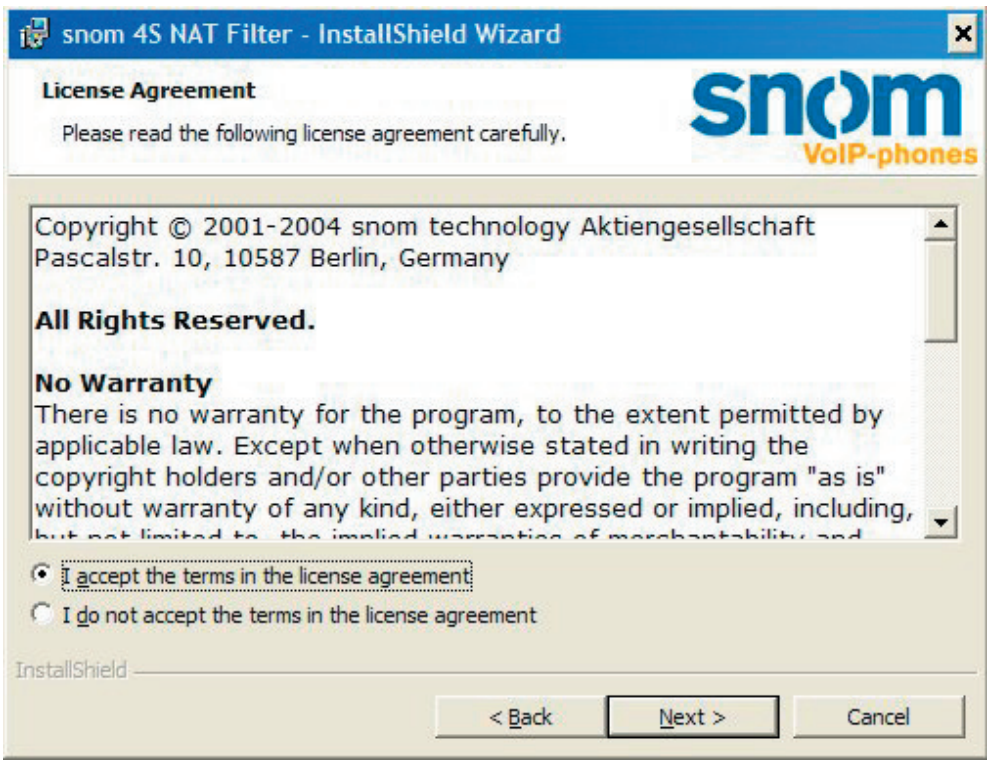
netstat command to check which ports are being used on that machine. You can change the ports later; however you should at least make sure that you can access the administration web interface of the NAT Filter with an open port.

Also, please make sure that you have the necessary administrator rights to run Windows services.

To start the installation, simply double-click on the installation executable. You will see the Welcome screen of the installation dialog.

To continue the installation read the text and click on the "Next"§ button. It will guide you to the license agreement page.

To continue the installation, please read the license agreement. If you agree with the license conditions, select the "I accept the terms in the license agreement" selection box and click on the Next button. If you do not agree with the conditions, you can cancel the installation by clicking on the "Cancel" button.



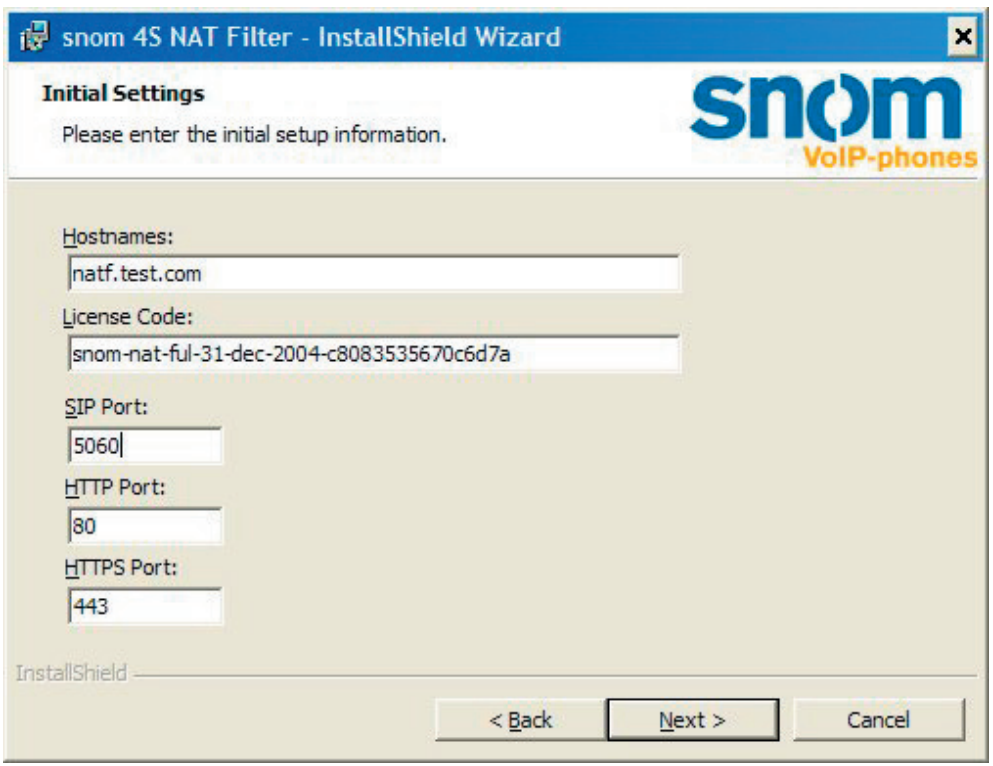
If you agree to the license agreement, the next screen will ask you to enter the license code and to select the ports of the NAT Filter.

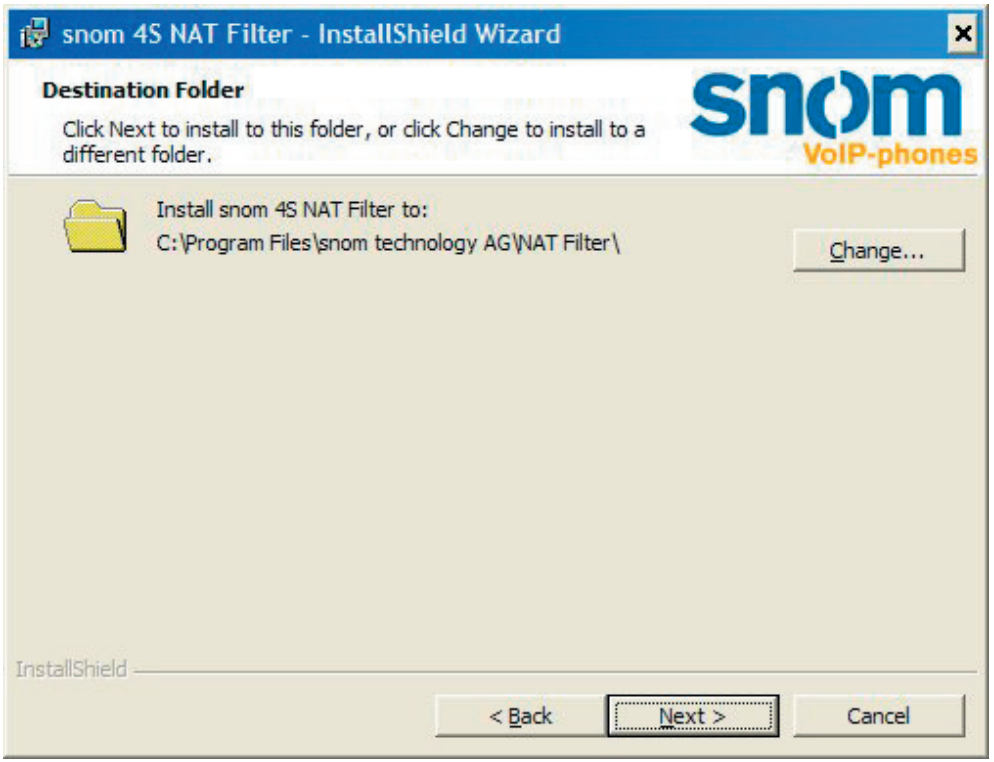
The hostnames are a list of host identifications that identify this installation. Typically, it is the list of DNS FQHN names for the used host.

You will receive the license code from the company where you bought the product. Please make sure that the code is correct (copy & paste).If you don't have a license key, NAT Filter will automatically generate a trial license key for you for a limited period of time. If you wish to use this mechanism, please leave the License Code field empty.

The SIP port will be used for communication of the NAT Filter with the outside world. The program will only open a UDP port, as other transport layers like TCP or TLS are not supported.

The http and the https ports are important for you as it is the only way to administer the NAT Filter. Please select a port number that suits your needs. The default ports are 80 (http) and 443 (https). If you





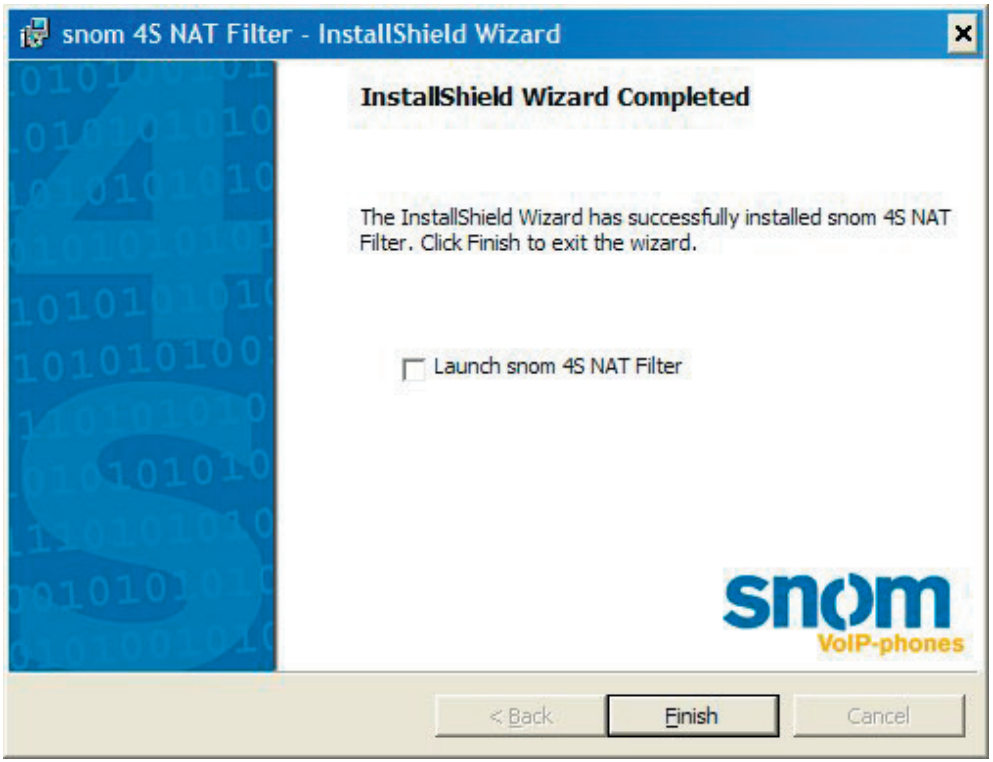
forget the port number, you need to look it up later, using the netstat command.

After entering the license information and the port numbers, the InstallShield program will ask you for the installation directory. Typically it proposes a reasonable directory; however you may change the directory using the Change button in this installation dialog.

After you have entered the necessary information, the last dialog will ask you to start the installation. You will see a progress indication. The installation typically takes only a few seconds.

The installation includes the registration of the NAT Filter with the Windows Services Manager. The next time when you reboot the machine, Windows will automatically start the NAT Filter. This is independent of a user login; this is important when you run the service on a server.

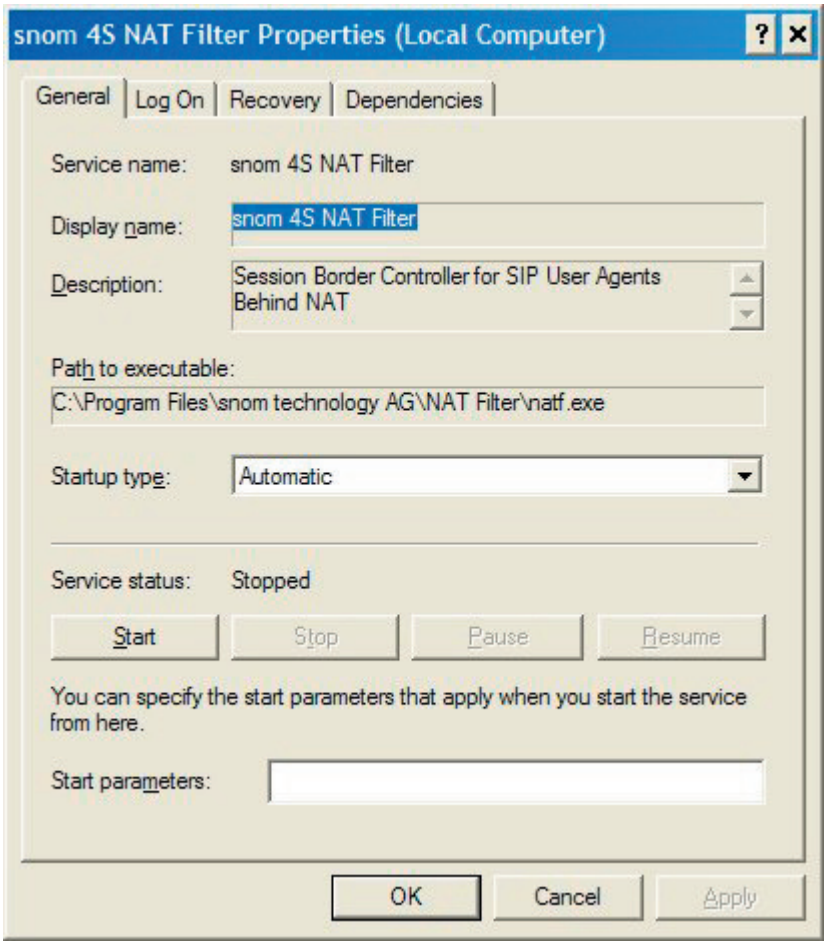
After the installation, the service is not started automatically. You can now either reboot to verify that the automatic start is working or you



may manually start the application using the services manager. The last InstallShield dialog offers you the option to start the NAT Filter. If you choose this option, you don't have to go to the services manager.

To see the NAT Filter service, go to the Control Panel, select "Administrative Tools" and double-click on "Services". You will see the list of services, including the snom 4S NAT Filter. If you select the properties menu entry, you will see the Properties dialog for the NAT Filter.

The NAT Filter does not require and start parameters, therefore you should not make any changes in the dialog. If you do not want the NAT Filter to start automatically, you can modify the Startup type to manual.



3.2 Linux

After you downloaded the RPM from our web site you can either install it via the graphical administration frontend of your Linux distribution or you can use the command line interface (CLI).

For the graphical installation please consult the documentation of your Linux distribution for details how to install 3rd party software.

If you use the CLI you need to be root to install the software. Please go the directory where you saved the RPM after downloading. If

this is the first installation of the snom 4S proxy on this host from a RPM package please use the following command to install the software:

```
rpm -ihv snomnatf-2.10.*.rpm
```

If you already installed an older RPM version of the proxy please use the following command instead:

```
rpm -Uhv snomnatf-2.10.*.rpm
```

The output of both commands will just show some hashes (#) and then return to the command prompt without any message if no error occurred.

After you installed the software please load the file /etc/sysconfig/snomnatf in your favorite editor and verify that you are satisfied with the default settings (SIP port: 5060, HTTP port: 80, Configuration directory: /var/lib/snomnatf).

Note: during the installation values from /etc/rc.config or /etc/natf.xml if they exists will be copied to /etc/sysconfig/snomnatf. The usage of /etc/rc.config or /etc/natf.xml is deprecated and only the values from /etc/sysconfig/snomnatf will be considered for the future.

When you are satisfied with the configuration values please start the proxy with the following command:

```
/etc/init.d/snomnatf start
```

Note: the process will not be started automatically any more like it was with the old snom tarball installation, because user interactions are not possible during a RPM installation, but the port settings should be verified by the user before starting the process.

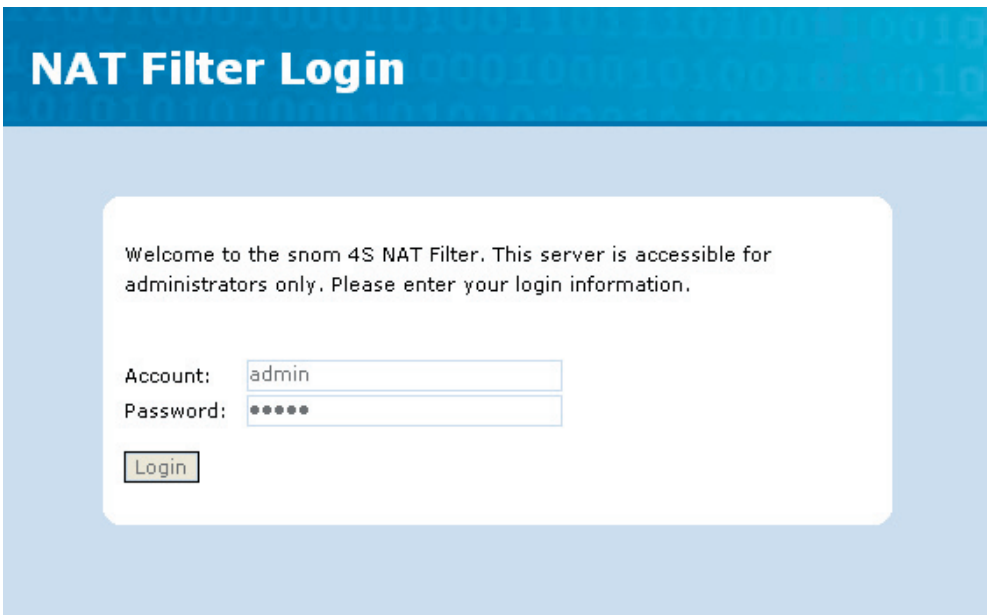
3.

4 Configuration

4.1 Logging In

First of all, you need to log in to the server (see figure 2). The default login name is "admin" and there is no password set (you should change this if it has not already been done for you).

The login creates a session. This session will timeout after a certain time (by default, one hour).



4.2 Port Binding

You need to tell the server on what ports it should listen.

For http and https, you need to know the port numbers when you want to log in. We recommend not using the standard ports. Operating a server on the public internet usually leads to a lot of denial of service attacks on the standard ports.

For sip, you must decide if you want to run the server on a standard port or a random port.

Standard Port	Random Port
• User Agents that don't support DNS SRV can automatically find the server • SIP-aware firewalls automatically take care about user agents behind NAT	• Buggy SIP-aware firewalls don't introduce new problems by modifying SIP packets • Less dangerous for DoS attacks • Several SIP services can be run on the same host

The decision depends on the situation. If you plan to use a good SIP firewall, you should choose the standard port. Otherwise we would tend to recommend a random port. Non NAT-aware user agents usually must be configured manually anyway; in this case you can also provide a port number.

The port for secure sip (sips) is usually 5061. The decision which port to use is similar to the decision for the SIP port. We recommend using a random port and publising the port number using DNS SRV.

In some situations when you have several IP addresses you want to limit the bindings to a specific IP address. You can do this by selecting the appropriate address from the pull down menu. If you choose „Default Address", the server will bind to all available addresses. If you select „Public Address", the server will select a public address; if you select „Private Address", the server will select a private address.

4.3 System Settings

4.3.1 Logging

The **Log Level** defines the granularity with which messages are written into the log. A log level 0 means that only the most urgent

messages are written, a log level of 9 means that all possible log messages are written.

If the **Log Filename** is set, all log messages are also written to the indicated file. If the file name contains a dollar character, the dollar will be replaced with the current date. Using this method, the NAT Filter will write a log file for every day. This way you can keep a certain history of log files and remove them from the file system as soon as you think the information contained there is not relevant any more.

The **Log Length** number indicates how many log entries the NAT Filter should keep in internal memory. The NAT Filter writes log messages using the first-in-first-out principle, so that there is no memory leak caused by log messages. The log messages written to the log file are not affected by this setting.

4.3.2 Preparing Recovery

You should specify a file name, so that the NAT Filter can **Save Registrations to File**. The filter will append an XML line for each registration that is being refreshed to a file that has the same name as the registration file appended with a tilde symbol (for example, if you specify "regs.xml", it will write it to "regs.xml~"). After the **Registration Logging Time** (see in timeout handling below, in seconds) the filter will move the tilde file to the main file. When the server is restarted it will read both files and this way restore the registration status on the filter. This allows the continuation of the service without waiting for the user agents to reregister. This interval should be longer than the maximum time that you give user agents for reregistration.

4.3.3 General Outound Proxy

The **Outbound Proxy** indicates where messages that are not coming from a UA behind NAT should be sent. Typically, this is the address of the proxy handling the traffic for the domain the NAT Filter is responsible for.

The outbound proxy is a SIP URI, which means it has the format sip:host. If the host contains a number behind a semicolon (as in "sip:proxy.com:5060", for example), the NAT Filter will just do a DNS A query on the address. If not, it will follow RFC3263 (Locating SIP Servers) to find the proxy. If you use DNS SRV, you can put a server farm behind the

NAT Filter. Because the NAT Filter itself can be operated in a server farm, you can set up a completely redundant server setup.

Please see also the list of explicit outbound proxies.

4.3.4 Media Ports

The **Media Port Begin** and **Media Port End** indicate the range of ports that are used for media relaying. Be sure to have enough ports allocated for the number of calls that you wish to route through the NAT Filter. This is a setting you may have to coordinate with your firewall.

4.3.5 Port Budgets

Because on Windows and Linux systems the number of TCP connections is limited, you can define budgets for the different kind of TCP connections.

The **Number of Web Connections** defines how many sockets for web connections (TCP and TLS) may be used, the **Number of SIP Connections** defines how many user agents may connect to the SBC at the same time. Please note that the underlying operating system defines the limits.

4.3.6 Media Relay

If you set the **Always Relay** flag, the filter will always relay media via the filter and will not allow bypassing it by ICE contacts. That means it will remove ICE contacts from the SDP and not insert an additional address for itself. This flag is useful when you want to make sure that all media flows through the filter, e.g. for measurement purposes or because you want to be able to record all calls. However, it will not be possible to do local media path optimization if you turn this flag on.

4.3.7 Controlling Routing

The **Loose Routing** flag influences the way the NAT Filter inserts routing headers into SIP packets. Loose routing is the routing mechanism proposed in the latest SIP document; however there are devices which are not able to deal properly with these routing headers (the new standard is not backward compatible with the old standard).

The **Hide Routing** flag will replace route sets with a unique route index when requests or responses are sent to a registered user agent. Via headers are also replaced with one Via header. This feature has several advantages. First of all, it will reduce the packet size significantly, especially when your core network uses several proxies or when it loops requests through the proxy several times. Usually, UDP packets will have a size significantly below the MTU size of 1492 bytes for Ethernet. This is a tremendous advantage that solves many problems with equipment that does not support UDP fragmentation.

Secondly, it hides important information about your network topology from the user agents. For example, when you are terminating calls with a PSTN gateway, the users are not able to see the IP address of the PSTN gateway in the routing path (if you turn "always relay" on, this address will also not occur in the SDP). Users will only "see" the filter as the only window to the outside world. This makes attacks much more difficult. It is much easier to protect only the filter against attacks than your whole SIP network.

The third big advantage is that it solves many problems with poor SIP implementations. Typically, immature SIP implementations cannot deal properly with strict and loose routing which results in complicated routing problems. The filter will take care of the routing problems; the user agent just has to route the request to the filter, which even the poorest implementations are able to do.

The disadvantage with this flag is that it adds more stateful information to the filter. The stateful does not affect the scalability of the overall system, but when restarting the filter, the information gets lost. However, we recommend turning this flag on.

4.3.8 Multiple 2xx Handling

The **Filter INVITE 2xx** deals with another problem that many poor SIP implementations have. In SIP, it is allowed to fork requests to several user agent servers. Several user agents sending a 2xx response back to the UAC at the same time typically creates a race condition. The proxy involved in this transaction cannot cancel the pending requests fast enough to solve this situation. The SIP designers have made the design decision that in this situation all 2xx responses must be sent back to the UAC which has to resolve the condition.

Unfortunately, only a small percentage of existing user agents deal properly with this situation. When you turn the flag on, the filter will only let the first 2xx response pass through to the user agent. Subsequent 2xx responses will be blocked by the filter; instead the filter will send an ACK to the response and immediately terminate the dialog with a BYE message. This is the behaviour of most user agents when receiving multiple 2xx. However, if you are sure that the user agents in your network handle multiple 2xx properly and implement a different behaviour, you should turn this behaviour off.

4.3.9 Challenging

Challenging inside a dialog may be problematic when the call destination does not have any credentials for the system. In this case, it may for example not be able to disconnect a call (BYE gets challenged). Therefore, the SBC may omit the challenging if the setting **Challenge Inside Dialog** is set to off.

Challenging every request may cause almost double packet traffic on the SBC for registrations. It gives you the maximum security, but in most situations it is reasonable to challenge only the requests that will be forwarded to the registrar. The setting **Challenge Refresh Registrations** controls this behaviour.

4.3.10 Trusted Addresses

The list of **Trusted IP Addresses** is used when sensitive information is extracted from SIP packets. For example, the filter may get an explicit hint on how long the conversation may last at most. If a user agent would send this information, it could easily bypass AAA and make telephone calls even when the prepaid card has expired. If you list the IP addresses of your proxies, you can enhance the security significantly.

4.3.11 Maximum Packet Size

The **Max MTU** tells the filter what the maximum packet size should be. Typically, on Ethernet networks, packets with more than 1492 bytes payload cannot be transported without splitting them up into several packets. As described in the hide routing feature, this can lead to big problems in today's DSL networks.

If you set this variable, the NAT filter will attempt to compress the message until it fits into the size. By default, it will use the short names (e.g. "l" instead of "Content-Length"). If this should not be enough, it will start to remove headers. These headers are: "User-Agent", "Accept-Language", "P-Key-Flags", "Allow", and "Allow-Events". If the packet is still too big, it will stop compressing the packet and send it as it is. If you want to remove other headers, please use the "remove the following headers" feature described below.

4.3.12 Silence Suppression

Silence suppression is a little problematic for the filter. When a user agent does not send media, it might lead to closing of allocated NAT ports on the media. Therefore, it is usually safer to turn silence suppression off. We recommend doing this by provisioning the respective setting to the user agents; however there is a way to indicate this in the SDP as well. If you turn the **Add silence suppression flag** on, the filter will add this hint to the SDP. Usually it does not cause any additional problems; however it makes the packet a little bit bigger which could cause additional problems with the UDP fragmentation problem.

4.3.13 Connection Oriented Media

Typically, you want two-way communication between the same ports in a conversation. Unfortunately, the IETF specifications do not mandate this. For example, it is allowed to have different ports for sending and for receiving data. This causes big problems when trying to make phone calls trough NAT. The comedia approach tries to standardize the requirements on using the same port for sending and receiving and to indicate if two-way communication is really desired. By turning the **Add comedia flag** feature on, you will make the filter add a suitable flag to SDP to indicate that this behaviour is desired. The disadvantage of this flag is again that it makes the messages bigger and this increases the probability that you will have problems with UDP fragmentation.

4.3.14 Removing Headers

As stated before, you may want to remove some headers to make messages shorter. The "**Remove the following headers"** setting lists the headers (separated by space) that you want to strip from a SIP

packet. This setting does not only help you in making the packets shorter, it can also help you to keep some parts of the SIP message secret. For example, you might want to remove P-Asserted-Identity headers from all SIP messages, because you don't want others to see which identities you already checked.

4.3.15 Codec Control

In many environments, you want to exclude codecs from being used, although both communication partners could agree on them. The "**Allow only the following codecs"** setting lists the codecs (separated by space) that you will allow. If you don't set anything here, all codecs will be allowed. The codecs must be written in their SDP name, for example "ulaw", "alaw", "gsm", "g729", "g723", etc.

This feature can be used, for example, to make sure that only low-rate codecs are being used. The user agents might otherwise agree on ulaw, which might lead to breaking voice if the bandwidth is not sufficient for a stream using ulaw.

4.3.16 Web Server Integration

The description of the web server integration follows in the next chapter.

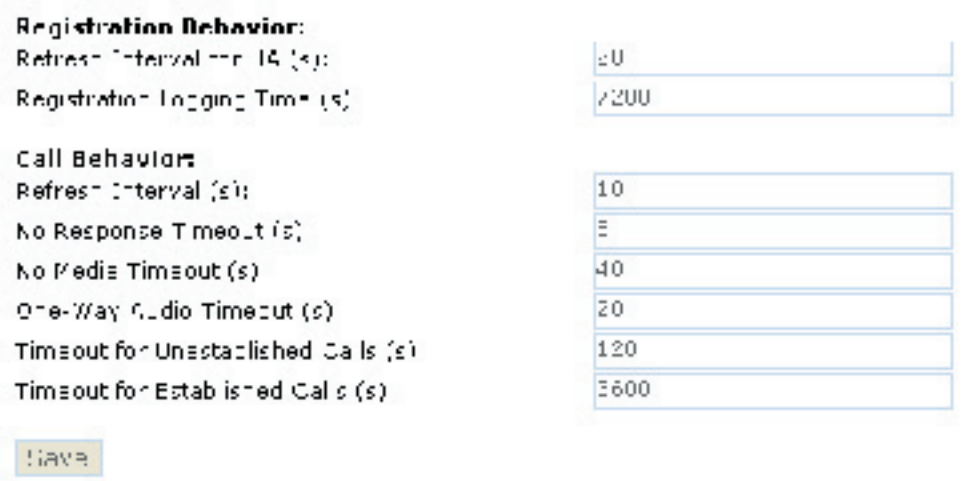
4.3.17 CLIR Addresses

The SBC has the possibility to hide the identity of the caller (see description in the following chapter). For this feature it must decide when to hide the identity. If it hides the identity too early, the further processing in the data center will be difficult. Therefore it must make a decision when the request leaves the influence area of the data center.

When the request is sent to a UA that is handled by the NAT Filter, it does perform this step. However, when it is sent to a PSTN or IP-gateway, it also must hide the identity. The setting Outbound Addresses (for CLIR) lists the IP addresses that also trigger a hiding of the identity. The format for this setting is the same as for the trusted IP addresses.

4.4 Timeout Settings

In contrast to previous versions, the time related settings have been summarized on this new management web page. The filter differentiates between registration related settings and call related settings.



4.4.1 Register Timeouts

The **Refresh Interval for the UA** is the number of seconds between NAT refreshes from the NAT Filter. The NAT Filter keeps track of registered user agents and keeps their NAT port binding alive with packets. Typically, a port timeout is 60 seconds; however, because this value must handle the smallest timeout, a value of 15 seconds seems more appropriate. The filter sends the keep-alive packets regardless of the transport layer. Although on TCP most implementations keep the connection alive for a much longer time, some implementations close their ports after a short timeout. Therefore, TCP connections must also be refreshed.

The **Registration Logging Time** is the time after which it moves the backup file to the primary location. See preparing recovery above.

4.4.2 Call Timeouts

Unfortunately, in SIP little attention has been given to the problem of a user agent disconnecting from the network without further notification. This situation typically occurs on power failure or system crash or when the Internet connection becomes unavailable. In these cases the filter needs to disconnect the call.

Even more unfortunate, there is no way this problem can be addressed. Therefore, the filter uses several mechanisms to check if the call is still alive.

The first way to find out if the call is still alive is to send OPTIONS requests to the user agents directly connected to the filter. The OPTIONS are sent outside of the dialog, because sending them inside the dialog would cause a sequence numbering problem. If no response comes back, it is an indication that the user agent is not connected any more (the reverse it not necessarily true: some user agents boot up so fast that options responses might be returned in time). The **Refresh Interval** tell the filter after how many seconds it should send; the **No Response Timeout** tells the filter how long it should wait for a response.

If there is absolutely no media, this is also a fairly good indication that the call is over. The setting **No Media Timeout** defines how many seconds the filter will tolerate calls without media. This setting only applies when any media has been received at all. This means if you, for example, had started an instant messaging session without any media, the filter will not remove this session because of a media timeout.

Another famous case is one way audio. Imagine a user agent calling a mailbox and then crashing/rebooting. The media server will play one-way audio possibly for a long time. Because of this scenario, we added the **One-Way Audio Timeout**. In contrast to the no media timeout, this setting only looks at audio media. This timeout is also only started upon reception of the first media packet. This setting should consider that some user agents do silence suppression. Please keep in mind that some users are very quiet and don't say anything for a relatively long time. However, a conversation where someone does not say anything for more than five minutes should be extremely seldom.

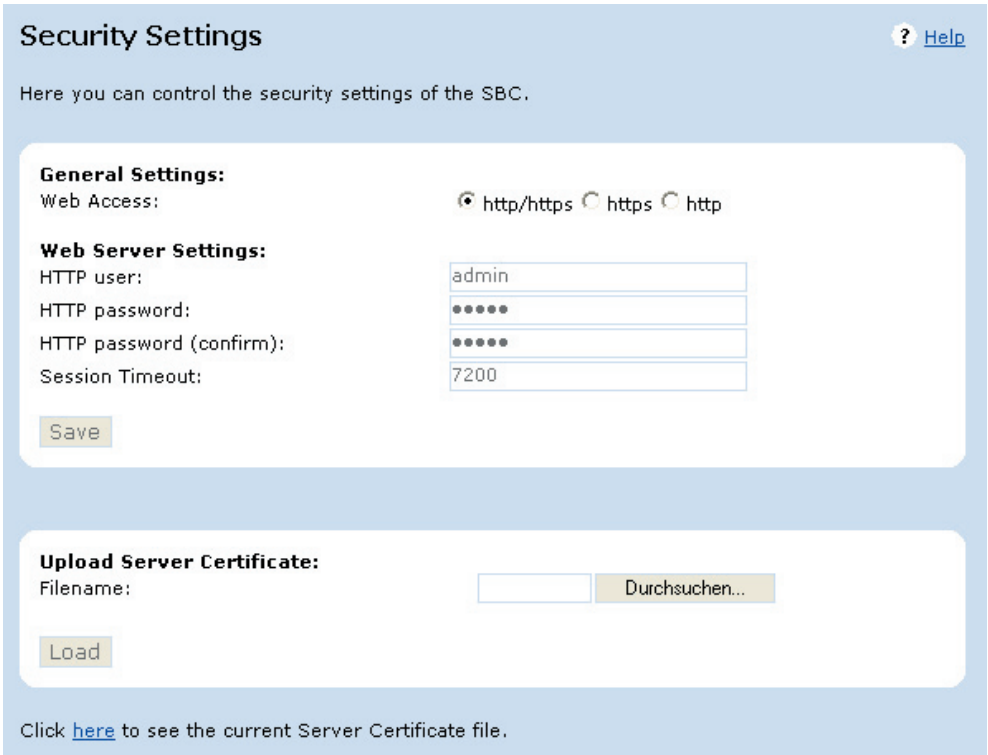
Another famous problem in SIP is to detect when the call is really over. In SIP, it is possible to challenge a request and wait for a relatively long time until the challenge is answered (for example, if the user has to answer the challenge by entering some data). If the challenged request

is not answered after a certain timeout, the filter assumes that the call is over and will not start again. The setting **Timeout for Unestablished Calls** addresses this problem. Please keep in mind that the ringing phone also falls into this category. Therefore, you should pick a value significantly higher than sixty seconds for this setting. On the other hand, every call attempt will stay in memory until this time is over. If you have many call attempts, you should keep this setting as low as possible. We believe that a value of 120 seconds is reasonable in most situations.

If the call has been accepted by receiving a 2xx on INVITE, it will probably last for a relatively long time without any signalling refreshes. Unfortunately, in SIP the session timer is not mandatory and has not been implemented in most of the user agents. Therefore, the filter cannot expect refresh traffic on the call. The setting **Timeout for Established Calls** defines the maximum duration that you accept without any SIP traffic. In other words, this is the maximum conversation time that you accept without any SIP traffic. The other timeouts do not stop the filter from hanging up because of this setting. Therefore, you should pick a reasonable time like four hours for this value. Typically this setting will only be needed for calls that don't have any media at all and where one of the user agents gets disconnected without notice. However, this setting is necessary to make sure that such calls do not fill the filter over time and reduce the number of available calls.

4.5 Security Settings

If you want to restrict access to the web interface of the NAT Filter, you may select the https radio button in the Web Access setting. If you don't care, you can leave the setting to http/https. If you want to exclude https access (for example, to save performance), select http.



To restrict the login, you should set a username („admin" is the default) and a password. You need to enter the password twice, so that typing mistakes do not block your NAT Filter.

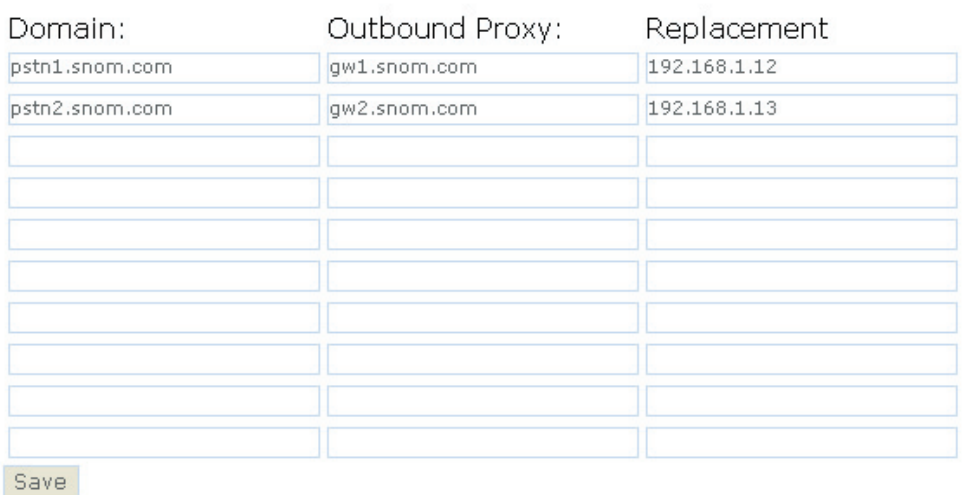
The Session Timeout is the number of seconds after which the NAT Filter web server deletes the session. If you access the web server after this time, you need to log on again. If you change the password during a session, you do not have to enter the new password for the existing session.

If you have bought a certificate, you may upload that certificate from the web page. Just point the NAT Filter to the file and press the Load button. Otherwise, the server will use a default certificate which causes you to trigger a security warning popup when you enter the web page. However, as the server is not open to public access, we think this is not so important.

4.6 Outbound Proxy List

In addition to the previously mentioned outbound proxy you may specify a number of dedicated outbound proxies. This feature is typically being used in the following scenarios:

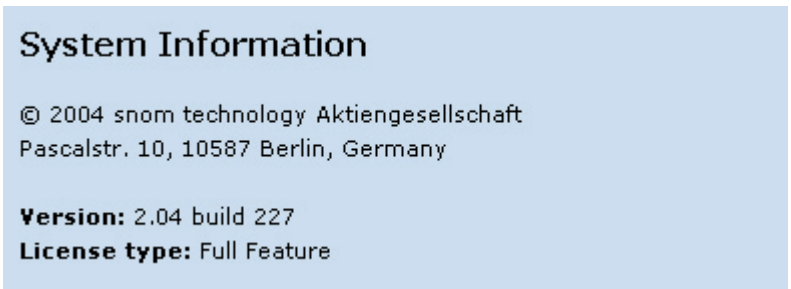
- Integration of PSTN gateway. If you set the outbound proxy of the PSTN gateway to the filter, it can easily redirect all requests to the proxy. However, when the proxy wants to route a call back through the filter, it must know that the request must be routed to the PSTN gateway. If you set up a DNS name for the PSTN gateway and set the destination to the filter, you can elegantly redirect all outgoing calls to the PSTN gateway trough the filter first. BTW the filter can then take care of such problems as UDP fragmentation and multiple 2xx responses.
- Replace PSTN gateway with other gateways, for example ITSP service or IP gateways.
- Multiple-Domain hosting: If you are using several proxies for different domains, the filter can redirect the requests to the right proxy automatically. This approach is limited to a maximum of ten domains per filter; for additional domains you need to use the web server integration mentioned above.



The algorithm for searching the outbound proxy is simple. The filter first goes through to the list of outbound proxies and tries to match the hostname in the request-URI of the request to the provided **Domain**. If it does not find a match, it will take the outbound proxy in the general settings (if provided). If it does find a match, it will replace the hostname part of the request-URI with the **Replacement** and then send it to the **Outbound Proxy**. Note that like in all routing decisions, the filter fully supports RFC3264; that means transport layer (tcp, tls, udp) and final destination are determined through DNS NAPTR, SRV and A lookups.

4.7 System Information

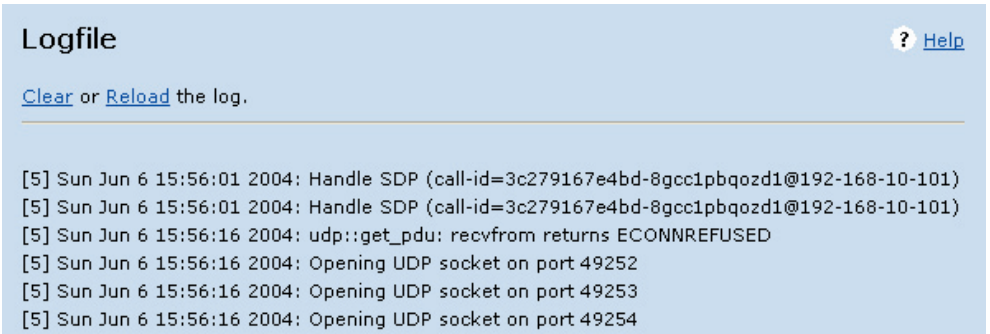
In the system information you can check the exact build number and the license type. We also include license information on this web page.



4.8 Server Log

This web page shows the log information that is kept in the memory (see above). It is independent from the logging information written to the log file.

You find two links that clear the log. Please remember that pressing the reload button on the web browser asks the NAT Filter to clear the log again. To refresh, you should press the link for refreshing the log. For your convenience, these links are available at the top and the bottom of the page.



4.9 Trace

The NAT Filter keeps a list of the last trace entries in memory. You may view this list by selecting the trace link. The handling of the page is similar to the handling of the log page.

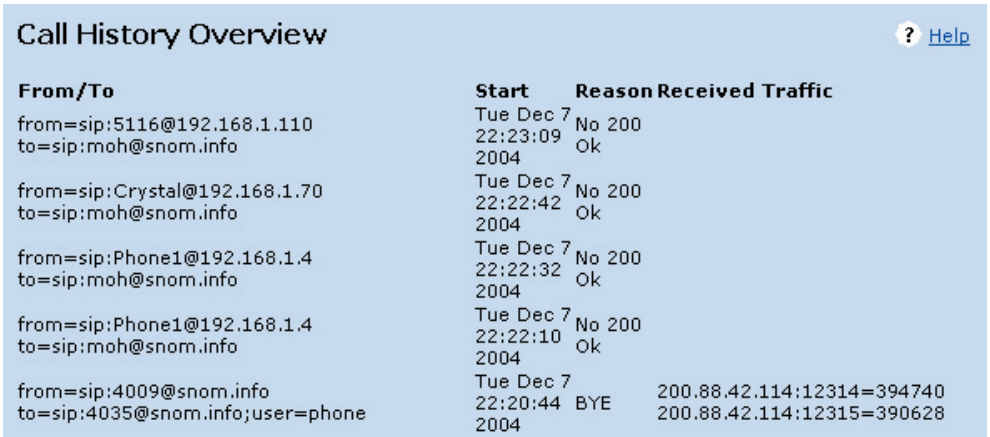
Each line contains an abstract of the received or sent packet. The Time column shows you when the packet has been sent or received.

The Type shows if the packet has been sent or received; in particular, „Tx" means the packet has been sent normally from the NAT Filter, "Tr" means the packet has been sent as message repetition, "Td" means the packet was sent to a UA behind NAT, "Rx" means the packet was received normally, "Rr" means the packet was received as a message repetition.

The Source/Destination indicates the IP address where the packet was sent or received. The Header column contains the abstract. By clicking on the header link, you may see the complete packet.

4.10 Call History

The call history should help you understand what's going on on your system. It is not intended to be an AAA feature.



The call history lists the last 32 calls. Each entry lists the to and from header (only the URI part). The start field shows when the call started with its first packet. This time is not identical with the time when the call was established, this is usually a little later.

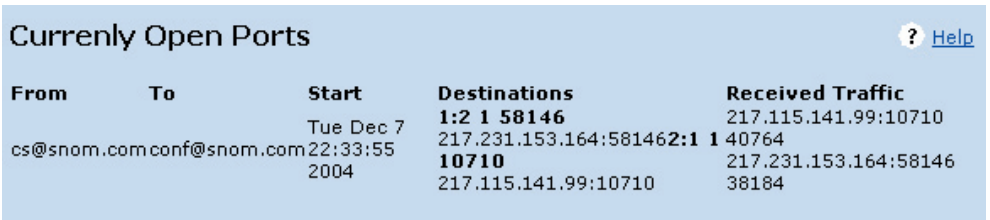
The reason field shows the reason why the call was terminated.

- The reason "BYE" indicates that the call was terminated because of explicit hang-up. This is the case for normal, successful calls.
- The reason "No 200 Ok" is typical when calls have not been picked up (call attempt).
- The reason "media timeout" indicates that the call was terminated because of a media timeout.
- The reason "OPTIONS" indicates that there was no response to an OPTIONS request.
- The reason "Maximum Session Duration" indicates that the session

was terminated because the maximum session time has been reached. This time is indicated by the P-Session-Timeout header.

4.11 Current Ports

It is important to see which calls are active on the filter. The Current Ports web page lists the calls where the filter performs relaying on media.



The from and to-field show which participants are involved in this media relay. The start column shows you when the port was created.

In the destinations field the user may see more information about how the different streams in the SDP are mapped. Each line consists of information about one stream. The number in bold before the stream shows the stream index. If that stream has been mapped to another stream, the number in bold behind the colon indicates what stream it has been mapped to. The number behind the space shows the index in the SDP. Because a conversation can have more than one SDP, the index usually occurs in several places. The indexes are matched by their value, according to their position in the SDP. The next number in bold shows the port number.

The next field shows the default destination that was indicated in the SDP. If the destination has not been locked, that address is shown in brackets and the list of learned addresses is shown after it. An address is locked when the NAT Filter received a packet on this port from the location indicated in the SDP.

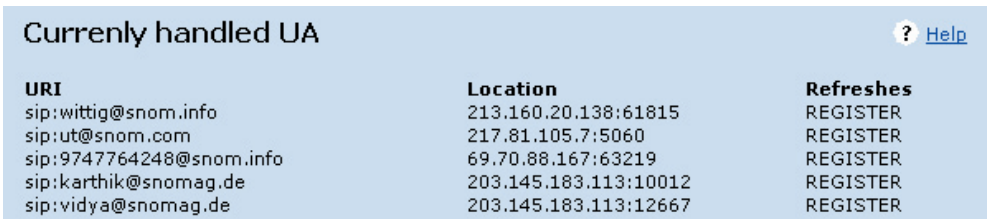
4.12 Currently Handled UA

This table shows the currently handled UA with their SIP URI and their associated IP address. The third column shows the SIP request type that this binding is using. Typically, this will be a REGISTER or SUBSCRIBE request.

User agents may have more than one entry. In this case, they might have dangling registrations on the registrar; this typically happens when the call-id is changed during a re-registration. From the NAT Filter point of view, this is no reason for concern.

It is ok if user agents show up several times. This typically indicates that the user agent tries to register several times, possibly on different proxies or after rebooting. The logic of the filter will make sure that only one refresh per destination occurs.

Please remember that calling this web page on a large installation would block the system for a potentially long time. Therefore, we limited the display to the first hundred registered user agents. You will see three dots at the bottom of the screen. If you want to check which user agents are registered on your system, you should check the registration log file instead.


Currenly handled UA

? Help

URI	Location	Refreshes
sip:wittig@snom.info	213.160.20.138:61815	REGISTER
sip:ut@snom.com	217.81.105.7:5060	REGISTER
sip:9747764248@snom.info	69.70.88.167:63219	REGISTER
sip:karthik@snomag.de	203.145.183.113:10012	REGISTER
sip:vidya@snomag.de	203.145.183.113:12667	REGISTER

4.13 Memory Statistics

This web page shows information about the current memory usage. The primary goal is to identify situations when the process grows more than expected. Usually, the NAT Filter process should not take more than five megabytes.

5. Web Server Integration

The SBC can use a web server as application server. This way you can use PHP, ASP and anything you like to implement the logic for your SIP traffic. For example, if you want to redirect a call to a specific gateway, you can do this easily on the web server. The SBC will just use the results that come from the web server to the further processing of the SIP request.

The SBC divides the interface to the application server into three areas:

- Authentication. Incoming requests are checked against a password list. If the user agent is not present in the local internal database, the SBC talks to the application server to update the information.
- Registration. When a user agent wants to register its contact, the SBC updates the registration information in the application server.
- Calls. When a new call is requested by a user agent, the SBC talks to the application server. The application server may route the call, change From and To-header, set the maximum call duration or reject the request.

The filter talks asynchronously to the web server. That means, the processing of other requests is not blocked by the request. Also the processing of other web requests is also not blocked by another web request. The request is only made at the beginning of a call; further messages inside the call will not cause additional web requests.

Please use the complete form for the server settings, including the "http://" in front of the host name. The filter supports only DNS A resolution for locating the web server; no http proxy is allowed.

5.1 Interface to the Web Server

The interface to the web server is built upon http. The communication is a request/response protocol. The SBC requests information from the application server, and the application server answers. The reverse communication direction is neither possible not necessary.

All requests are formulated as GET requests. The parameters are URL-encoded. The typical request has the form:

```
GET /post.htm?action=register&from=123@test.com HTTP/1.1
Host: test.com:8181
Accept-Language: en-us
Connection: Keep-Alive
Keep-Alive: 5
User-Agent: Mozilla/4.0 (compatible; snom)
```

The responses contain the answer in the body. The SBC checks the response code, and if the code is 2xx, it processes the attachment.

The attachment is encoded using a simple line-based protocol.

```
From: „Albert Einstein“ <sip:albert@einstein>
To: „Isaac Newton“ <sip:isaac@newton>
```

The name of the header is printed before a colon. It is not case sensitive. The argument follows the colon. If there is white space before or at the end, this white space is stripped. Therefore, the two headers are equivalent:

```
From: „Albert Einstein“ <sip:albert@einstein>
from:“Albert Einstein“ <sip:albert@einstein>
```

Note that the white space inside the argument is not changed.

The SBC uses only http (TCP), a secure transport layer is not supported. Also, UDP or other transport layers are not supported. The current version opens a new TCP connection for every request. This may be optimized in future versions.

5.2 Authentication

If the Http URL for registration is set in the system settings, the SBC performs the following algorithm for every request:

- If the packet was already authenticated or internally generated, the further processing of the packet can start.
- If the request is a register request and the registration is still valid, the packet forwarded to the further processing. This behaviour can be disabled with the "Challenge Refresh Registrations" setting.
- If the packet belongs to an existing call and is not the initial INVITE, the packet is forwarded to the further processing. This behaviour can be disabled with the "Challenge Inside Dialog" setting.
- If the packet comes from a trusted IP address, the following checks are performed. If the request comes directly from a UA (there is exactly one Via header), the packet is forwarded to the further processing. In this case the SBC will insert a P-Asserted-Identity header. If the packet contains more than one Via-header, the packet is only forwarded to the further processing, if the P-Asserted-Identity header is already present. In this case, the SBC will overwrite the header with the present value of the From-header.
- If the request method is ACK or CANCEL, the packet is forwarded to the further processing. Note that in this case the SBC does not insert a P-Asserted-Identity header.
- The SBC then looks at the user and host part of the From-header of the request URI. If that pair is not present in the authentication cache, it requests that pair from the application server and stops processing the request until the answer is available. If during this request more messages arrive for the same user/host pair, these requests are queued until the answer from the application server is available. When the answer from the applications server request is available, the packet is processed from the beginning of this algorithm again.
- If the user/host pair is present in the authentication cache, the SBC will check if the packet contains the correct answer to a challenge. Note, that typically during the first time of processing a request this is not the case and the packet gets challenged with a new allocated nonce. If this check succeeds, the SBC adds a P-Asserted-Identity header to the request and forwards it for further processing.
- Otherwise, it will allocate a new nonce and challenge the request. The nonce represents a question that can only be answered by the shared secret, the password of that user/host pair. The nonce will expire after one hour and is deleted when the question is answered

correctly.

The web requests that the SBC sends to the application server has the following parameters:

- The parameter "action" is set to "auth". By looking at this parameter, the application server can easily find out that it should do a password lookup.
- The parameter "from" contains the user/host pain. It has the format user@host, there is no scheme and no parameters included in this parameter.

The authentication cache is written with every web response. The response may contain any number of answered, but must at least contain the requests user/host pair. The answer must be encoded in a comma separated value format with no header line. The CSV response has the following fields:

- The first cell contains the user/host pair in the same format as in the request. The SBC identifies the cache entry by this cell.
- The second cell contains the user name that should be used for the challenging. Typically, this is identical with the user name part of the from cell, but sometimes the challenging should use a different name.
- The third cell contains the realm that should be used for the challenging. Again, this field should typically be identical to the host part of the from header, but in some situations this realm can be different (for example, when canonical realm names must be used).
- The fourth cell contains the password in clear text.
- The fifth cell contains the expiration of that cache entry in seconds. After this time the SBC will remove the entry from the list and issue another applications server request to refresh the values. A typical value would be one hour (3600 seconds).

The SBC interprets the presence of the parameters in the following way:

- If the realm or the username are not set, that user will always be challenged with no hope for recovery. That practically means that the request is denied.
- If realm and username are set but the password is empty, the request will not be challenged; instead it will be assumed that the user

is authenticated.

- If realm, username and password are set, the request is regularly processed.

Because it is possible to send more than credential with one authentication request, the applications server can update passwords that have just been changed. By using this "piggyback" method, changed can be propagated into the SBC relatively quickly after the user changed his or her password.

5.3 Registration

If the "Http URL for registration" setting is set in the system settings and a register request does not refresh an existing binding, the SBC sends a request to the application server with the following parameters.

- The parameter "action" is set to "register". By looking at this parameter, the application server can easily find out that it should do a registration.
- The parameter "from" is set to the user/host pair of the From-header. The encoding is in the user@host format.
- The parameter "contact" is set to the contact that represents the binding in the SBC. This parameter has the From/To-Spec format of RFC3261. Typically, the parameter will look like "<sip:1.2.3.4:5060 ;ua=345a20f784c9284>". Note that the parameter will be URL-encoded, which converts special characters into the respective representation.
- The parameter "expires" contains the proposed expiry time from the registration request of the user agent.

Registration refreshes do not trigger application server interaction. This way the load on the application server can be kept reasonable low. Note that the authentication step is performed before the registration actions.

The applications server must return a response with the following parameters:

- The parameter "code" contains the SIP response code for the request. If the registration is ok, this typically will be a 200. If the user does not exist in the registrar, the code will typically be 404.

- The parameter "explanation" contains the explaining text that is added behind the code in the SIP response. Typical values are "Ok" or "Not Found".
- The parameter "contact" contains the contact that should be returned by the registration response. This parameter contains the expiry time that this contact will have. Typical values look like "<sip :1.2.3.4:5060;ua=345a20f784c9284>;expires=3600". This parameter may be present more than once if the application server has several bindings. If there is no binding, the parameter can be omitted.

If the UA does not reregister, the SBC will send a register request with an expiry time of zero seconds. This emulates the deregistration request of the user agent.

5.4 Call Initiation

If the "Http URL for call" setting is set in the system settings, the SBC will consult the application server for every call that is being initiated.

A new call is identified by an unknown Call-ID and starts with an INVITE request. The SBC will perform the authentication steps before the call is initiated (see above). For a new call, the request from the SBC contains the following parameters:

- The parameter "action" is set to "start". By looking at this parameter, the application server can easily find out that it should do a call start.
- The parameter "from_uri" contains the URI part of the From-header. Typically, this has the form "sip:albert@einstein". Note that the port number and parameters may also be present in this parameter.
- The parameter "to_uri" contains the URI part of the To-header like the "from_uri" parameter.
- The parameter "uri" contains the request-URI.
- The parameter "callid" contains the Call-ID of the call.
- The parameter "from_ua" is set to "true" if the SBC believes that the call comes from a client endpoint.

- The parameter "to_ua" is set to "true" if the SBC believes that the call will go to a client endpoint. Note that this may change during the processing of the request.

The SBC expects responses with the following parameters (line-encoded like register responses):

- The parameter "from" contains the value of the From-header as is should look like. The SBC will change the From-header accordingly, however it will leave the parameters of the From-header unchanged (for example, the tag). You may include the display name and the URI as you like. This way, you can use canonical names for the account name or insert the display name from the address book.
- Similar, you can return a "to" parameter.
- If you return the "uri" parameter, the SBC will route the call to the provided SIP URI. A typical value looks like "sip:123@proxy.com;parm=1234". You may include any number of parameters. This is useful for passing additional information to the next element in the routing process. For example, you can pass parameters to a proxy that describes to which destinations it should fork the call and after which time. Also note that according to RFC3261 you may include headers in the URI which will be expanded into packet headers. For example, if you return the line "uri: sip:123@test.com?MyHeader=Hello", the request URI will become "sip:123@test.com" and the header with the name "MyHeader" will be inserted into the packet with the content "Hello".
- The parameter with the name "id" will carry a token that will be returned on the termination web request. With this variable, you can match the start and end transactions.
- If you set the parameter "expires" to a number, you limit the maximum duration of that call. The value contains the number of seconds.
- If you return a number if the parameter "error_code", the request will be rejected with that code. The parameter "error_message" will contain the text that is used as error explanation (for example "Not Allowed"). Please don't use error codes below 300 (see the SIP RFC if you are not sure which error code to take).
- If the parameter "anonymous" is present, the SBC will insert a Privacy header into the request. When the request leaves the data cen-

ter, the From-header will be set to the value that you pass here.

Please note that requests may loop through several SBC. This will typically happen in data centres that use a SBC server farm. In this environment, the application server must be able to handle several call initiation requests for the same call as the SBC do not exchange information about web requests. A simple implementation just passes an empty response. The subsequent SBC will then just route that call without any other change of the request.

When the uri parameter is present, the SBC does not use the outbound proxies to route the request. You must make sure in the application logic that an appropriate request URI is inserted.

5.5 Call Termination

When the SBC calls the destructor for the call object, it first sends out web request that informs the applications server that the call has finished. In contrast to the other web requests the answer to that request does not matter — the call is finished anyway.

The SBC provides the following arguments to the request:

- The parameter "action" is set to "end".
- If the id parameter was set in the response to the start request, the SBC will put that argument in the parameter "id". The application server may use this information to match the call.
- The parameter "callid" is set to the Call-ID of the call.
- The parameters "from_uri" and "to_uri" are set to the same values as they had during the action=start call.
- The parameter "reason" is set to the message that describes why the desructor was called (see below).
- There are several time values available. All these times are measured in seconds since Jan 1st, 1970. The parameter "time_invite" indicates when the initial INVITE was received. The parameter "time_180" indicates when a 18x request was received (if there was such a response), the parameter "time_200" indicates when the first 2xx code was received. The "time_bye" parameter (if present) shows when a BYE request was received (this parameter is only present when a 2xx was received).

Currently, the following reasons are available:

- "BYE" means that the call was terminated by a regular BYE message.
- "No 200 Ok" is used when the call did not establish (4xx code or other final error codes).
- "OPTIONS" is used when the call was terminated because there was no response to an OPTIONS refresh request of the SBC.
- "media timeout" indicates that the call was terminated because the SBC detected that the media flows was disrupted.
- "one-way audio" is similar to media timeout, but happens when there was a longer period of one way audio.
- "Maximum Session Duration" is used when the SBC closes the call because the maximum duration (provided in the "expires" parameter in the action=start result) has been reached.

5.

6. SNMP

The simple network management protocol (SNMP) is a widely used protocol for checking what's going on in your network. When you run the SBC, you probably also want to see statistics about the usage and get alarms when something goes wrong.

6.1 Setup of the SBC

The setup of SNMP on the SBC side is very simple. Essentially, you have to perform two steps:

- Select the port on which the SNMP server should listen. By default, this would be port 161, but on a host that runs other SNMP services as well you might want to choose another port. This step must be done in the Port Binding web page.
- Tell the SBC from which addresses to accept SNMP requests. In the Security Settings web page, you find the setting "Trusted IP Addresses" in the SNMP section. Enter the IP addresses (separated by a space) or the IP address range here. The SBC will accept requests only from these addresses.

The IP addresses must be in the form dots-and-number notation. The SBC does not perform a DNS resolution of the addresses. If you want to specify a range of addresses use the form Adr/Bits, where bits is a number indicating how many bits of the IP address should be considered. For example, the string "192.168.2.0/24" would match addresses 192.168.2.0 until 192.168.2.255.

6.2 Setup of the Tools

The setup of the tools varies from tool to tool. Because the SBC does not offer a standard set of values (such as CPU temperature, disk

space etc.), the setup is a little bit more difficult that the setup of a standard sensor.

A readable parameter is described by its object identifier (OID). The object identifies are described in the next paragraph. Please enter the OID in your tool and select appropriate names for them. Also make sure that the IP address of the host running the SNMP tool matches the setup that you gave the SBC in the "Trusted IP Addresses" setting.

The SBC does not support "snmpwalk" or other tools that automatically describe the abilities of the SBC. You must enter these settings manually.

6.3. Available OID

The following table describes the available OID. An absolute value describes the current state on the SBC, the value might go up and down. Relative values only go up and accumulate the values.

OID	Description	Absolute	Unit
1.3.6.1.2.1.7526.1.1	Open Calls	Yes	Calls
1.3.6.1.2.1.7526.1.2	Registrations	Yes	Registrations
1.3.6.1.2.1.7526.1.3	Minutes	No	Minutes
1.3.6.1.2.1.7526.1.4	Received Media	No	Bytes
1.3.6.1.2.1.7526.1.5	SIP Packets	No	Packets
1.3.6.1.2.1.7526.1.6	SIP Traffic	No	Bytes
1.3.6.1.2.1.7526.1.7	Successful Calls	No	Calls
1.3.6.1.2.1.7526.1.8	Unsuccessful Calls	No	Calls

The minutes are measured only for the codecs ulaw, alaw, G.726 (32 kbit/s), GSM 6.10, G.723.1 and G.729. The minute measurement is done by looking at the media type in the RTP packet. If the packet uses a non-standard mapping, the results may be inaccurate.

The SBC measures only the received traffic. Usually, this is equal to the sent traffic, because the SBC does not block traffic. However, when there is a problem with the media establishment, the number of transmitted bytes might differ from the number of received bytes. The information does not include IP header like the UDP header.

For the SIP packets, the SBC measure only the received packets. It measures the number of packets as well as the total number of bytes received on the SBC SIP ports. The information does not include IP header like the UDP header.

The number of successful and unsuccessful calls is incremented after the call has finished. The counting corresponds to the logging of the call result which is described in a different part of this document.

6.

7 Checklist for Installation

When snom or one of their partners perform the installation for you, the following information is necessary:

6.1 Linux

- Please provide secure shell login to the system that can be accessed at least from the snom.com host (currently at IP address 217.115.141.99).
- Please tell us the login address (host and port), user name and password. We need root permissions on that host.
- Please tell us for which domains you plan to use the server. Please also tell us where you want to process the requests (which outbound proxy to use for NAT Filter).
- Please don't run too many other services on the host that can degrade the performance of the server. We recommend using the server only for NAT Filter.
- Please tell us which ports you want to use for NAT Filter (SIP, http, https). We recommend not using the standards ports, as they sometimes trigger inappropriate actions on immature equipment.
- Please set your DNS SRV records for SIP up so that they point to the NAT Filter SIP UDP port. Please don't use other transport layers than UDP. Do not set up DNS SRV records for TCP or TLS.

6.2 Windows

- Please make sure that we can access the host from the Internet. Please ask the installation team if they support a remote administra-

tion tool.

- Please tell us the login address (host and port), user name and password. We need administrative rights on that host.
- Please tell us for which domains you plan to use the server. Please also tell us where you want to process the requests (which outbound proxy to use for NAT Filter).
- Please don't run too many other services on the host that can degrade the performance of the server. We recommend using the server only for NAT Filter.
- Please tell us which ports you want to use for NAT Filter (SIP, http, https). We recommend not using the standards ports, as they sometimes trigger inappropriate actions on immature equipment.
- Please set your DNS SRV records for SIP up so that they point to the NAT Filter SIP UDP port. Please don't use other transport layers than UDP. Do not set up DNS SRV records for TCP or TLS.

References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, „SIP: Session Initiation Protocol", IETF, RFC 3261, June 2002, http://ietf.org.

[2] Rosenberg, J., Weinberger, J., Huitema, C., Mahy, R., „STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", IETF, RFC3489, March 2003, http://ietf.org.

[3] UPnP™ Forum, http://www.upnp.org/

[4] Rosenberg, J.: "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Multimedia Session Establishment Protocols" (Internet draft), http://ietf.org/internet-drafts/draft-ietf-mmusic-ice-02.txt

[5] Rosenberg, J., Mahy, R., Huitema, C.: "Traversal Using Relay NAT (TURN)" (Internet draft), http://www.ietf.org/internet-drafts/draft-rosenberg-midcom-turn-05.txt

Reader's Feedback

snom technology AG welcomes your evaluation of this manual and any suggestions you may have. These help us to improve the quality and usefulness of our documentation.

Please send your comments and suggestions to:

snom technology AG
Attention: Marketing Department
Pascalstr. 10B, 10587 Berlin, Germany
Fax: +49 (30) 39833-111

Manual Name: snom 4S NAT Filter Admin Manual 2.10

	Excellent	Good	Fair	Poor
How would you rate the document overall?	☐	☐	☐	☐
Are the installation instructions effective?	☐	☐	☐	☐
Are the configuration instructions effective?	☐	☐	☐	☐
Is the document properly organized?	☐	☐	☐	☐
Are the illustrations usefull and easy to understand?	☐	☐	☐	☐
Are the suggested and default values useful?	☐	☐	☐	☐

Did you find any errors in the document (please reference page)? _____

_____

_____

_____

_____

How might we improve this manual? _____

_____

Name _____ Title _____

Company _____Telephone (____) _____

Thank you for taking time to fill out this form.

snom technology Aktiengesellschaft
Gradestr. 46, 12347 Berlin, Germany
Phone: +49 (30) 39833-0
mailto:info@snom.com
http://www.snom.com
sip:info@snom.com

snom
VoIP-phones